Microcontrollers

Errata Sheet

20 January 2003 / Release 1.2

Device:	**C868-1SR/SG**
Stepping Code / Marking:	**AA-0204 (AB)**
Package:	**P-TSSOP-38**
	P-DSO-28

This Errata Sheet describes the deviations from the current user documentation. The classification and numbering system is module oriented in a continual ascending sequence over several derivatives, as well already solved deviations are included. So gaps inside this enumeration could occur.

The current documentation is: User's Manual (0.6.1)
Data Sheet (0.6)
Instruction Set Manual 07.2000

Note: ***Devices marked with EES- or ES are engineering samples which may not be completely tested in all functional and electrical characteristics, therefore they should be used for evaluation only.***

The specific test conditions for EES and ES are documented in a separate Status Sheet.

Change summary to last Errata Sheet Rel. 1.1:

- Added new item numbered PD.1.

Functional Problems:

CCU.1: CCU interrupts do not participate in the arbitration of MCU and might be masked by other interrupts.

CCU interrupts do not go through the SCU module. The SCU has the function to lengthen the interrupts so that they can participate in the arbitration of the MCU. The SCU would keep the interrupts to the MCU active until the interrupt flags are reset. However, the CCU module does not offer this feature for CCU interrupts. Thus CCU interrupts are not able to participate in the arbitration of the MCU and might be masked by other interrupts.

Workaround:

To avoid CCU interrupts being masked by other interrupts, they must be set to high priority (priority 1) and all other interrupts of the same priority as CCU interrupts must check for the CCU interrupt flags. Suppose CCU interrupt 0, 1 and 2 together with external interrupt 0 are invoked in the application. CCU interrupt 0, 1 and 2 sources are CC0, CC1 and CC2 inputs respectively (falling edge triggered).

The following is a software workaround:

```
        ORG    0000H
        LJMP   MAIN

        ORG    0003H            ; external interrupt  0 address
        …                       ; external interrupt  0 routine itself
        MOV    A, ISL           ; check CCU interrupt flags
        JNZ    CCU
        RETI

        ORG   0083H             ; CCU interrupt  0 address
        SJMP  CCU

        ORG   008BH             ; CCU interrupt  1 address
        SJMP  CCU

        ORG   0093H             ; CCU interrupt  2 address
CCU:
        MOV   A, ISL
        JNB   ACC.1, CCU1       ; check CCU interrupt 0 flag
```

```
            …                       ; CCU interrupt  0 routine
CCU1:
            JNB   ACC.3, CCU2       ; check CCU interrupt 1 flag
            …                       ; CCU interrupt  1 routine
CCU2:
            JNB   ACC.5, CCUEND     ; check CCU interrupt 2 flag
            …                       ; CCU interrupt  2 routine
CCUEND:
            ORL   SYSCON0, #10H     ; set RMAP
            MOV   ISRL, #0FFH       ; reset CCU interrupt flags
            ANL   SYSCON0, #0EFH    ; clear RMAP
            MOV   A, ISL
            JNZ   CCU
            RETI

            ORG   0100H             ; main program
            …
            MOV   IP0, #00111100B   ; set CCU interrupts to high priority (priority 1)
            …
            …
```

CCU.2: T12 Shadow Transfer for Phase Delay Function

In Hall mode (T12MSELx = '1001'), T12 never reaches its period value in normal operation because a detected hall event on CCPOSx captures the actual T12 count value to CC60R (speed reference) and **resets T12**. But this period event is needed to trigger the shadow transfer of T12 register, i.e. an update of the timer registers (e.g. CC61R as variable phase delay) is not possible. The shadow transfer should be triggered together with the reset event of T12.

Workaround:

The shadow registers of T12 are transparent if the timer is stopped and if the shadow transfer is enabled (STE12=1). Therefore it is possible to write the new value to the shadow registers by enabling the shadow transfer, stopping T12 and starting T12 again. This is equivalent to a hardware triggered shadow transfer event. Inevitably, some clocks for the capture/compare events are lost in between the stop / start operation. It has to be noted that bit STE12 is cleared by hardware after a shadow transfer event.

Application Hint:

WDT.1: Watchdog Timer (WDT) status in Bootstrap mode.

The Watchdog Timer (WDT) has been disabled in the bootstrap mode and cannot be enabled when boot-up from Bootstrap mode. Thus the WDT does not run and does not generate a WDT reset.

PD.1: After wakeup from power down PLL might unlock if oscillator startup is long.

In the User's Manual, it is stated that there will a period of start-up phase before PLL starts locking. Conversely, PLL starts to lock even during the start-up phase. And once the lock detection circuit has output an 'OK' signal, the program will start running. However, due to instability of start-up phase and inadequate lock detection period of 4096 clocks (153 us @ 10.67 MHz), PLL might unlock and generate a PLL-reset. This will cause a rebooting if BSL mode is enabled and therefore the wakeup process is corrupted.

Workaround:

Use a ceramic resonator instead of a crystal with a fast oscillator startup phase.

Deviation from Electrical- and Timing Specification:

The following deviations of electrical and timing parameters from the specification are known in this step:

None.

History List (since last CPU step AA-0150)

Functional Problems

Functional Problem	Short Description	Fixed
CCU.1	CCU interrupts do not participate in the arbitration of MCU and might be masked by other interrupts.	
CCU.2	T12 Shadow Transfer for Phase Delay Function.	
IPD.1	Ipd (power-down current) is too high at hybrid pad supply power.	AA-0204
WDT.1	Watchdog Timer (WDT) status in the Bootstrap mode.	Hint
PD.1	After wakeup from power down PLL might unlock if oscillator startup is long.	Hint

AC/DC Deviations

AC/DC Deviation	Short Description	Fixed
None.		

Application Support Group, Singapore